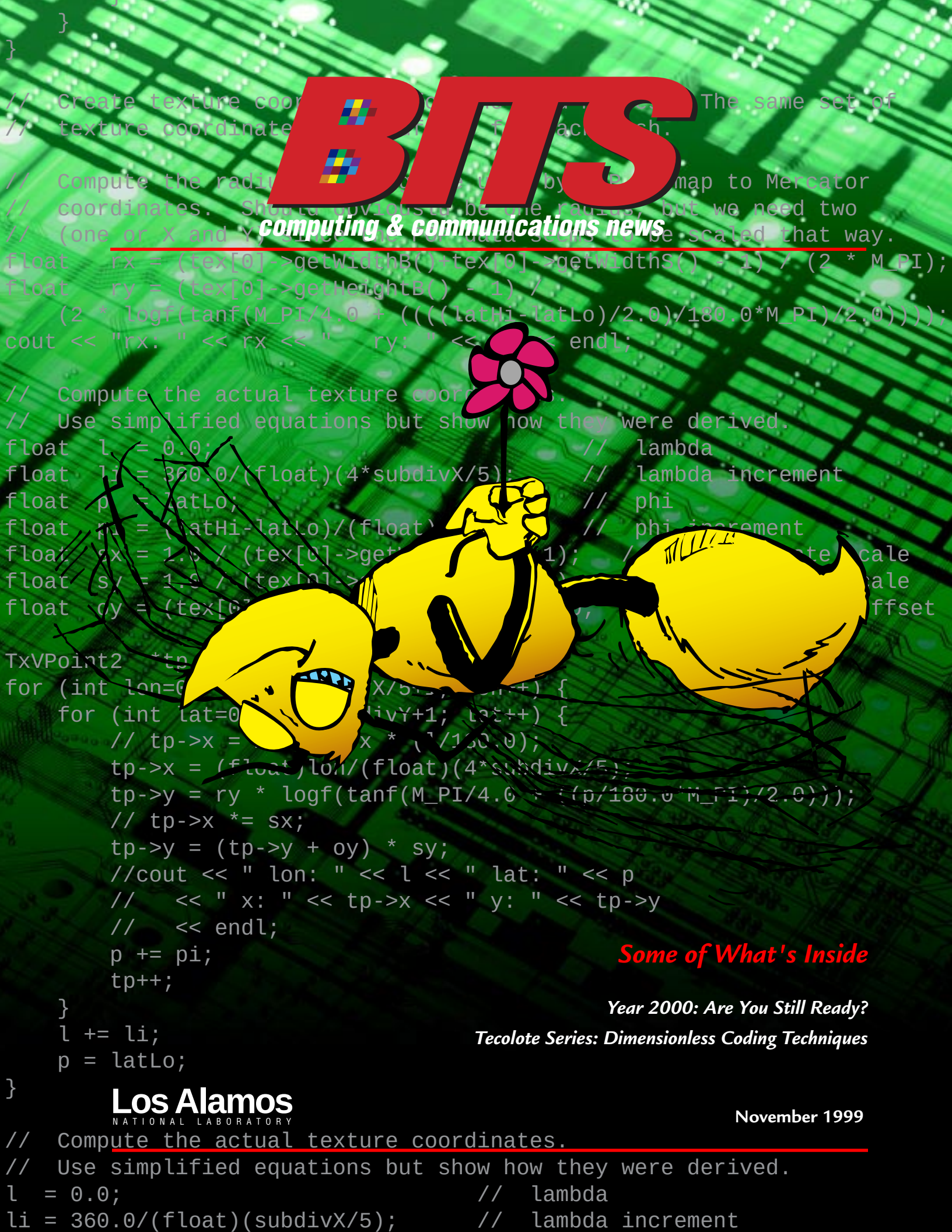
BITS

computing & communications news



Some of What's Inside

Year 2000: Are You Still Ready?

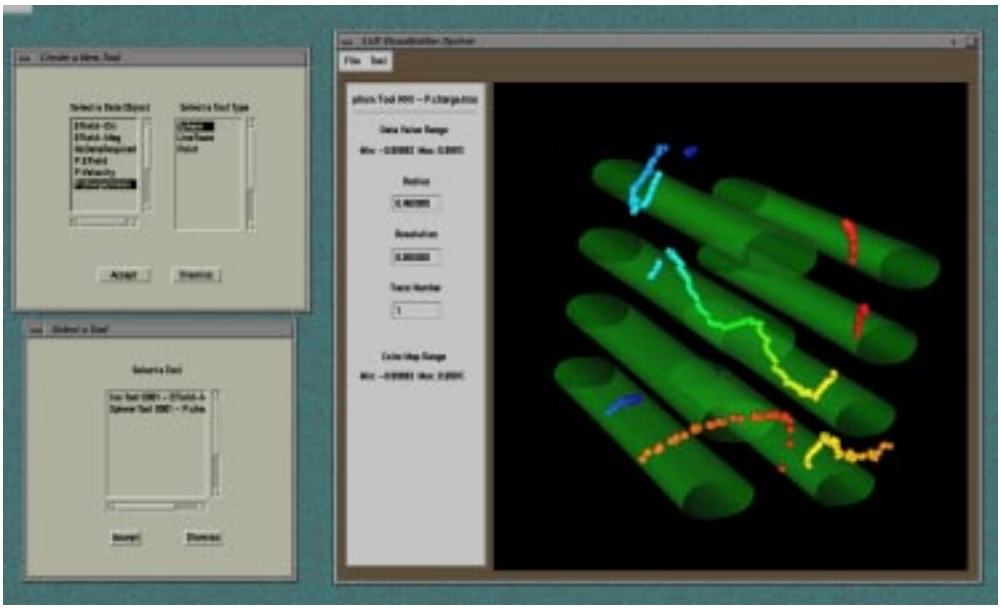
Tecolote Series: Dimensionless Coding Techniques

Los Alamos
NATIONAL LABORATORY

November 1999

LUX Visualization System

This figure shows a visualization of a particle-in-cell calculation. The figure shows an isosurface of an electric field and the location of charged particles. This visualization was created using the LUX visualization tool. Project Leader-Jim Ahrens. For more information contact the Advanced Computing Laboratory's Visualization Team at viz@acl.lanl.gov.



Produced by the Computing, Information, and Communications (CIC) Division
Design: Donald Montoya and Julie Medina (CIC-1)
Managing Editor: Denise Sessions, denise@lanl.gov (CIC-1)
Illustration: Dave Delano-Art/Illustration
Printing: Imaging Services Group (CIC-9)

Attention Subscribers: There are two ways you can update your subscription data. E-mail bitsupdate@lanl.gov, or fill out the form inside this issue. Internal LANL subscribers please note that until we have a mechanism in place to verify our subscription database with the LANL Employee Information System we will rely on you to update your mailstop. Thanks for keeping us updated on subscription data.

Table of Contents

Information Systems **2**
Scalable Vector Graphics—Web Graphics with Original-Quality Artwork 2

Desktop Computing **7**
ADSM: A LANL Centralized, Automated Backup and Restore Option 7

Infrastructure **11**
Take a Look at the New PAGES 11
Recharge System Telecommunications Enhancements 13
Year 2000: Are You Still Ready? 14

High Performance Computing **18**
Complex Systems Modeling: Using Metaphors from Nature in Simulation and Scientific Models 18
Dimensionless Coding Techniques in Tecolote: Using a Single Source-Code Base for Multidimension Programs 22

What's Happening **26**
Book Review 26
New Interface for Biosciences Database 27
Beilstein Organic Chemistry Database Is Now on the Web—Free 27
Research Library Resources 28
Software Documentation Services 28
Technical and Advanced Computer Training Courses 29
BITS Subscription/Comment Form 30
1999 12-Month Index 31

Scalable Vector Graphics—Web Graphics with Original-Quality Artwork

by Tad Lane, Information Architecture Standards Editor, CIC-1 Communication Arts and Services

Ever wanted to shrink or zoom a Web graphic without distortion? How about size it so it takes up the same 15% of a page width regardless of the screen resolution of the browser viewing it? Pan it to the side? How about zooming way far in on a map to see just the part you want to see...without distortion?

Traditional ways of doing any of the above on the Web are clumsy, at best. The main details you see when you zoom in on current inline graphics formats are the details of their limitations—smudginess for JPEGs, blockiness for GIFs or PNGs. (If you're not familiar with these acronyms, please see the Acronym sidebar in this article.) The traditional workaround has involved multiple images—click on part of a map and you wait for a whole new image to load to show that detail.

With Scalable Vector Graphics (SVG), however, we can do all of the above, and more, with resolution to the limit of our monitor. We can use the full truecolor colorspace; we can include text; we can use event handlers (onmouseover, onclick, etc.); we can create animations through scripts; we can even insert traditional JPEG or PNG images if we want to. Or, more accurately, we will be able to do all those things as support for SVG becomes more widespread.

Current Status

SVG, like the MathML discussed in last month's BITS article, is an eXtensible Markup Language (XML) language that focuses in on a specific Web need—the need for two-dimensional vector graphics. (See below for a further explanation of vector graphics.) As of this writing, SVG is a World Wide Web Consortium (W3C) Working Draft, currently in last call, which means we can expect some minor changes before it becomes a W3C Proposed Recommendation.

There are several Java applications already available for viewing SVG, including applets that can be associated with the SVG file itself via traditional HTML markup. These are usable (I used the CSIRO Mathematical and Information Sciences SVG viewer for the examples in this article), but they require the preloading of up-to-date Java and an XML parser. Also, because the specification itself is not yet fully stable, these should be considered experimental demonstrations rather than production applications.

The near future looks bright, though. Adobe, Apple, Corel, IBM, Sun, Macromedia, Netscape, and Microsoft are all members of the W3C SVG Working Group, and all have voiced support for the specification. Hence, within the next year, it seems reasonable to expect familiar commercial applications for both creating and rendering SVG.

Image Format Basics

There are two basic types of electronic image formats: raster formats, which describe an image as an array of pixels (dots); and vector formats, which describe an image as a group of mathematically defined shapes (lines, circles, etc.). Traditional GIF, JPEG, and PNG images are all raster graphics, which are still useful in certain cases. (PNG is rapidly displacing GIF for Web graphics, and I'll talk about the future for the rest of this article by not mentioning GIF again.)

Most of the applications we use to create Web graphics internally use vector formats. When we create a shape in a program such as Photoshop or Fireworks, it's the underlying vector format that allows us to move the shape around, resize it, apply textures, distort it, zoom in without losing resolution, and so on. Traditionally, we've created high quality graphics in these tools and then exported them to the lower quality raster formats. What SVG gives us is the ability to export the graphics to a vector format that retains all of the quality of the original artwork.

Traditionally, the way we've put vector graphics on the Web has generally been through PDF or PostScript formats. The advantages of SVG over these formats are that it's targeted to graphics, lighter weight, vendor neutral, and tightly integrated with other Web specifications through XML. Under HTML, SVG like PDF needs to be stored in a separate file; when we get the broader implementation of XML, though, SVG graphics can be integrated into an overall XML document. (Note that the W3C has also issued a profile for "WebCGM" (Computer

Graphics Metafile) that targets vector graphics for engineering and medical industries, but that is beyond the scope of this article.)

Vector graphics are not the best choice for all images. For photographs, for example, JPEG remains the better option. For images such as screenshots that are already reduced to pixels, PNG is generally the better option. For images such as computer-generated line art, though, SVG is the better choice, especially if we want to do things like scaling the image.

For an example of scaling differences, later in this article I show SVG markup to create a logo for the Information Architecture (IA) Project. (I chose the logo because we're limited to black and white in the print version of this article.) Compare the following screenshots taken when zooming in near the dot of the "i" in the logo (see Figs. 1 and 2).

Fig. 1. Enlarged PNG with "Blockiness"

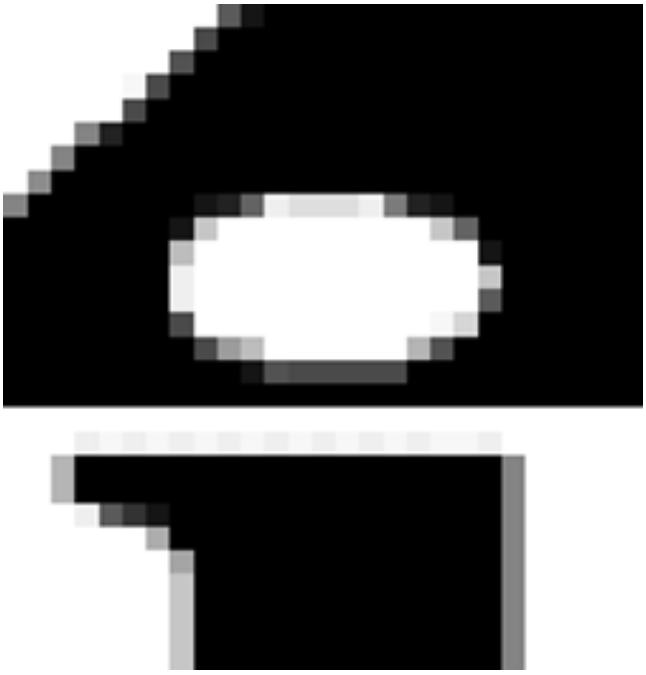


Fig. 2. Enlarged SVG with Sharp Edges

SVG compression also works differently from traditional image formats. For JPEG and PNG, the compression is applied to the image file itself. This is because JPEG and PNG define an image as an array of pixels of varying colors, and each uses specialized compression algorithms that are optimized for their particular uses. (JPEG, in fact, is a compression algorithm rather than an image format. JFIF is the real name for the JPEG File Interchange Format, even though we continue to say "JPEG" more commonly.)

SVG, however, like other XML languages, is text based. Hence, compression is applied not to the file itself but to the transmission of the file. HTTP/1.1, the current HyperText Transfer Protocol, applies compression that is optimized for text-based files.

such as HTMLs, XHTMLs, and SVGs.

SVG Basics

For anyone familiar with HTML or XML markup, SVG markup will likewise look familiar. Like HTML and XML, SVG consists of tags and attributes. For example, the following file defines a filled red circle:

```
<?xml version="1.0" standalone="no"?>

<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG August 1999//
EN"

"http://www.w3.org/Graphics/SVG/svg-
19990812.dtd">

<svg width="300" height="300">

<circle cx="150" cy="150" r="100" style="fill: red"
/>

</svg>
```

The document, like other XML documents, begins by declaring itself an XML document and by identifying its Document Type Definition (DTD). <svg>...</svg> define the beginning and end of the SVG content, with the intrinsic width and height defined as attributes of <svg>. (The intrinsic size is the size that the SVG image is defined in; the resulting SVG image can still be scaled to other sizes.)

The <circle /> tag defines the circle itself:

- "cx" gives the position of the center on the x axis (horizontal).
- "cy" gives the position of the center on the y axis (vertical).
- "r" gives the radius of the circle.
- "style" uses standard Cascading Style Sheet (CSS) conventions to define the color to fill the circle with.
- "/>" closes the tag per the XML requirements.

SVG can do more than just circles, of course. Although a comprehensive review of the specification is beyond the scope of this article, a look at some selected tags might give a sense of the capabilities of the format. (Note that examples that close with "/>" are empty, while the other tags have content and require closing tags, as in "<text>content</text>".)

- <rect/> defines rectangles by their position, height, width, and optional rounding of corners.
- <ellipse/> defines ellipses by their position, x radius, and y radius.
- <line/> defines lines by their starting and ending points.
- <polygon/> defines a closed polygon by the position of its various points.
- <polyline/> defines an open polyline by the position of its various points.
- <text> defines text by its position and content.
- <path/> defines curved lines including Bezier curves and elliptical arcs.
- "stroke" and "fill" are style properties that define the colors and patterns to apply to any of the above shapes, with stroke including characteristics such as width and linecaps.
- <linearGradient> and <radialGradient> define gradiated colors to use with strokes or fills.
- "onmouseover" and "onmouseclick" are two of many event handlers that can be used to activate scripts from parts of the image.
- <a> assigns hotlinks to parts of the image.

And those are just selected examples. There's also markup for filter effects, patterns, opacity, animation, external style sheets and more. Through combining and ordering the various pieces, we can create a broad range of Web graphics, roughly comparable to what we can natively create in programs such as PhotoShop or Fireworks today.

Remember, however, the major vendors who have been involved in the development of SVG. Their commitment to SVG means that we will probably see the format incorporated into user-friendly tools that will shield us from the underlying code (though some of us, as with HTML, will always want to get into the code to optimize it, tweak it, etc.).

Example IA Logo

As a simple example of how SVG works, I used SVG to create an IA "house" logo with the following markup. Note that I use a local copy of the DTD in this case, that each object as it is created is layered on top of the previous objects, and that comments are inserted within "<!— —>" the same way as in HTML and XML languages.

```
<?xml version="1.0" standalone="no"?>

<!DOCTYPE svg SYSTEM "svg-19990812.dtd">

<svg width="120" height="120">
```

```
<!— define the outside border as a black square
 with a smaller white square on top of it —>

<rect x="1" y="1" width="120" height="120"
 style="fill: black"/>

<rect x="10" y="10" width="102" height="102"
 style="fill: white"/>

<!— position the "ia" near the center of the image
 —>

<text style="font-size: 70; font-family: serif;
 font-weight: bolder; color: black" x="28"
 y="85">ia</text>

<!— build a black triangle that covers the dot of
 the "i" and a black rectangle for the base. note
 that the "g" tag groups the two objects and
 applies the black fill to both objects at the
 same time —>

<g style="fill: black">

<polygon points="60 12 106 51 14 51 60 12" />

<rect x="14" y="87" width="92" height="19" />

</g>

<!— create the white dot for the "i" —>

<ellipse cx="40" cy="44" rx="7" ry="4"
 style="fill:white" />

</svg>
```

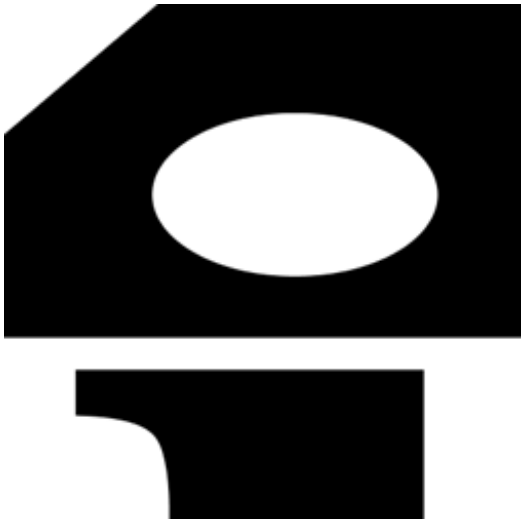


Fig. 3. Screenshot of SVG Image

A screenshot of the resulting image is shown in Fig. 3, as rendered on a CSIRO SVG Viewer.

Although this is a very simple graphic, it is interesting to note the difference in filesizes for different versions of the logo, with each version having the same dimensions (and with me breaking my promise to not mention GIF anymore):

- GIF logo - 2,420 bytes
- PNG logo - 1,541 bytes
- SVG logo - 550 bytes (with comments and extraneous white space removed)

On top of the already apparent differences, the GIF and PNG versions have already been compressed, while the SVG version doesn't get compressed until transmission.

For Further Information

For further information about SVG, see the W3C SVG page at http://www.w3.org/Graphics/SVG/ . This includes links to SVG applications such as the CSIRO SVG Viewer used for this article.

For further information about graphics formats, see the IA White Paper IA-6801: Electronic Image Formats and Compression Algorithms at http://www.lanl.gov/projects/ia/stds/ia680120.html . Although the paper needs to be updated to reflect SVG's emergence, it does cover formats such as PNG and JPEG in more detail, and it provides examples of PNG's superiority over GIF.

The following BITS articles have dealt with related topics:

- "MathML: A Kind of 'TeX for the Web' " (Oct 1999)
- "Web Content Architecture: Taming the Tangle of Protocols" (Sep 1999)
- "It's Time for PNG: A Graphics Format We Can Pronounce" (Oct-Nov 1998)
- "Extending Web Documents: Getting Ready for XML" (Mar 1998)
- "Images on the Web: More Tips" (Sep 1996)
- "Images on the Web: Some Tips" (Aug 1996)

For additional information about the IA Web activities, please see our Web team page at http://www.lanl.gov/projects/ia-lanl/area/web/ . For additional information about the IA Project in general, please see our project home page at http://www.lanl.gov/projects/ia/ .

Acronyms and Such

CGM–Computer Graphics Metafile

CSS—Cascading Style Sheets

DTD—Document Type Definition

GIF—Graphical Image Format

HTML—HyperText Markup Language

HTTP—HyperText Transfer Protocol

IA—Information Architecture Project

JFIF—JPEG File Interchange Format

JPEG—Joint Photographic Experts Group

MathML—Mathematical Markup Language

PDF—Portable Document Format

PNG—Portable Network Graphics

SVG—Scalable Vector Graphics

W3C—World Wide Web Consortium

XHTML—Extensible HTML

XML—eXtensible Markup Language

ADSM: A LANL Centralized, Automated Backup and Restore Option

by John Bremer, Storage Solutions Team Leader, CIC-7 Computing Group

Second Article in a Three-Part Series

Many people who lose data can revert to backup files. For individuals who fail to back up important documents, they risk being forced to recreate them should an "accident" occur. Imagine coming into work one morning to find everything on your computer is gone: no applications, no files, no Word documents, and no projects that you've been working on for the last couple of months. Gone, and you don't have your data backed up!

IBM's Adstar Distributed Storage Manager (ADSM) is a file storage and backup service available in both the unclassified/open and classified/secure networks at LANL. The CIC-7 Computing Group recommends using ADSM because it provides reliable, efficient, and secure automated file backup of workstations, personal computers and servers, and has an archival file storage capability for large files. Currently nearly 170 million files and 28 terabytes of data are stored in ADSM. Of these data, nearly 5 terabytes of data per month move between ADSM and its respective clients, via regular back up or restores. These data are currently managed across five ADSM servers in the open network and one server in the secure network. This article will inform readers about ADSM characteristics and the ADSM Team plans for implementing new initiatives, policies, and procedures.

ADSM is a good choice for backing up, archiving, restoring, and retrieving in preparation for and as an option for being able to continue to do work after the year-2000 rollover. It is advisable to back up your data onto some other media besides your hard drive so that you can conduct business in case of temporary loss of infrastructure as a result of the millennium change. ADSM can provide a backup option to help users proactively deal with any Y2K problems, such as loss of data on local personal computers.

Security

All ADSM servers reside in the Central Computing Facility (CCF), and only system administrators have access to each server. These accounts are password protected and require kerberos authentication for login. All ADSM servers are located on the Integrated Computing Network backbone, which consists entirely of switched, point-to-point, Fibre Distributed Data Interface (FDDI) circuits, which guarantees no "sniffing" or "hacking" of traffic to or from ADSM.

Access to ADSM data via the client machine is protected by an encrypted ADSM password (encrypted on the client machine) that is stored on the ADSM server. All user data are compressed on the client before transmission to the ADSM server, making the data unusable to anyone other than the ADSM client.

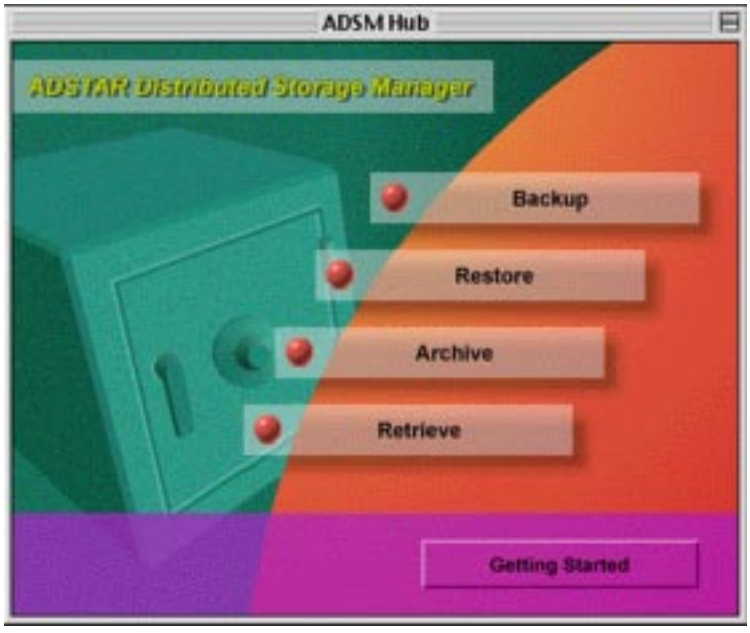


Fig. 1. Graphical user interface for ADSM software.

ADSM clients are set up so that a small number of invalid login attempts will cause the node to be locked. This would prevent any unauthorized attempt to determine the password of the node from malicious attacks. In addition the ADSM activity log includes the IP address of every start-session attempt, so tracing of malicious attacks can be tracked and acted upon; in other words, a client masquerading as another user would be detected because it would originate from a different IP address.

In the open network, ADSM is approved for storing unclassified protected and unprotected information up to and including Unclassified Controlled Nuclear Information, providing the user is behind the Yellow protected/open firewall. Sensitive unclassified data, such as personnel data, is certainly within the unclassified protected realm.

Last, all ADSM tape and disk media, before it leaves the CCF, is degaussed in NSA-approved and NSA-certified tape and disk degaussers, preventing the inadvertent release of information outside of the Laboratory. We are also exploring the possibility of placing an ADSM server outside the firewall, provided sufficient client machines in the Green unprotected/open network would benefit from having ADSM backup there.

ADSM Growth and Usage

In October 1997, there were 1328 registered users of ADSM at LANL; in October 1998, 2126 users; and currently, as of September 1999, 2447 users. This represents 60% growth in ADSM users in FY98, and 15% growth in the FY99. Meanwhile, the number of files managed by ADSM in FY99 has grown from 115 million to 185 million (61% growth), and total data stored from 17 terabytes to 28 terabytes (65% growth). Again, the critical factors for ADSM are not so much total data stored, because we have adequate tape capacity, but rather the growth in numbers of clients and files per server.

ADSM Storage Characteristics

The optimal number of clients per ADSM server is approximately 500; however, this number can vary greatly depending on the number of files, or ADSM objects, that each server is required to manage. Because ADSM is a back-up-by-file system, an ADSM server must manage each client file (object) in a random access database or meta-database. Because ADSM data storage itself is accommodated in automated tape libraries, actual data storage capacity far exceeds the capacity of each ADSM server to manage the meta-data efficiently. Therefore, the maximum number of clients an ADSM server can effectively handle is directly related to the number of files backed up by those clients.

In addition to this upper limit on the number of files/objects an ADSM server handles efficiently, there are also a threshold number of clients that can back up concurrently during normal backup windows. Because most clients opt for automatic backup between 6 p.m. and 6 a.m., perhaps as many as 450–500 clients per server will queue up to run during that time.

New Initiatives

With a configuration of five ADSM servers, we are reaching current capacity. To accommodate future growth and to handle it responsibly, the ADSM team is acquiring additional hardware, including new technology servers, disk and tape drives. Recently, one ADSM server grew to double the sustainable capacity (over 1000 clients) because of lack of resources to place a new server in production service. This severely impacted attempts by users to establish sessions with ADSM for backup during the normal ADSM automatic backup windows.

Current analysis of the 2447 existing ADSM client registrations shows that 13% of these accounts (334 clients) have never accessed ADSM, and 30% (731 clients) have been inactive (not backed up) in the last 30 days. The ADSM team is in the process of changing/adding procedures to reduce the number of these dormant clients.

Never Accessed and Dormant Accounts

Our strategy is to discover why the never-accessed accounts have not used the service (e.g., either because of installation problems, double registrations, lack of knowledge, etc.), and to determine why the inactive accounts remain dormant for such long periods. Actions we are taking to this end are

- delete all accounts that are no longer viable,
- make house-calls where necessary,
- recognize that some “inactive” accounts may be backing up every 30–60 days for a purpose (e.g., the data are not volatile),
- send e-mail inquiries to
  - — report the status of individual accounts,
  - — ask users to indicate the nature of their problems with ADSM (if any), and
  - — ask whether they wish to continue to use the service.

New Procedures and Policies

We've implemented these new procedures and policies to carry out our new initiatives:

- new interactive Web options for account management,
- audit of tape volumes and load balancing,
- upgrade Version 2 clients, and
- e-mail notification for backup failure.

New Interactive Web Options

Our Web options have been extended beyond general information about the service. Individuals may now modify many of their ADSM options without having to call or e-mail ADSM Help. For example, users can use the Web interface to query their most current backup status. (See http://storage.lanl.gov/cgi-bin/adsm/qfilesp.) We've implemented a Web-based ADSM User Account Update page with token card passcode access. Options include the following updates:

- secondary e-mail notification,
- account codes,
- backup window time,
- automatic e-mail notification, and
- delete obsolete ADSM client filespaces (see http://storage/cgi-bin/adsm/userupdate).

Audit Procedure and Load Balancing

To ensure that tape media and files are not damaged, we have implemented a continuous audit procedure that mounts and checks our inventory of ADSM tapes. Also, to prevent excessive growth on any one ADSM server, we have implemented a policy to load-balance the servers and restrict registrations when server capacity is not available, thus minimizing the possibility that a client might not get a session during their designated backup window.

Fig. 2. LANL's ADSM Web site home page.

Upgrade Version 2 Clients

Last, we are striving to bring our Version 2 clients up to current, ADSM-supported levels, where possible, and to notify others that our support of Version 2 will discontinue. Until now, our support of Version 2 clients has been "best effort", because IBM will no longer assist us with debugging Version 2 problems. For systems where no Version 2 client exists (e.g., SunOS) we can recommend options, such as backups via a Version 3 agent.

E-Mail Notifications for Backup Failures

At the inception of ADSM at LANL, users could choose among various intervals—daily, weekly, monthly, or never—to receive periodic e-mail notifications when their backups failed. In conjunction with new backup monitoring procedures noted above, we will be implementing a new policy to "notify never" or "notify every 30 days" in case of backup failure. In place of more frequent e-mail notification, we will provide more capabilities for clients to check the status of their latest backup, whether through scripts that will query the client logs to check last backup status, or through Web-based tools. This is more reliable and involves the client and user. Our philosophy is to involve and educate the users that it is also their responsibility to check backup status with the tools we have provided.

ADSM 2000

During October 1999, the ADSM team will begin implementing the new procedures discussed above. Other initiatives not already mentioned include

- new installation wizards for Mac and Windows clients that incorporate both registration and installation procedures into one process,
- design modifications and additions to the ADSM Web site, including up-to-date change control notifications (e.g., for new client versions), and
- new on-line help tutorials (audio/visual) for basic ADSM functions, such as installation, incremental backup, file restore, etc.

We recognize that installing and managing the ADSM client is not easy for everyone. When necessary, it is suggested that a qualified system administrator assist users with installations and problems; the ADSM storage team is also available at all times for customer assistance. We are working with the CIC-6 consultants to discover how to minimize specific problem tasks and probable causes, and to remedy problem areas when they are identified. The ADSM team will be concentrating on making the service easier to use and more responsive, and on providing better education about this product for our customers.

For more information about this article, or about ADSM in general, please contact John Bremer at 665-7209 or e-mail jbremer@lanl.gov.

Take a Look at the New PAGES

by Denise Sessions, BITS Managing Editor, CIC-1 Communication Arts & Services and Steve Hench, PAGES99 Project Leader, CIC-9 Imaging Services

PAGES (Print And Graphics Express Station) is operated by CIC-9. PAGES serves as the digital imaging and media output node of the Laboratory's Integrated Computer Network (ICN) in both the Open and Secure partitions. PAGES provides electronic shipping and image rendering tools necessary to create professional, publication-quality, computer generated reports and graphics via a wide array of media formats.

Meeting New Demands

Today's demands on PAGES are significantly different than those of the 1980s when the system was introduced. While the amount of plain-paper black-and-white printing is lower, the amount of data handled has grown dramatically. This is because text files, such as source code and output listings, can generate a lot of paper for only a few hundred kilobytes of data. Meanwhile, a single-page PostScript® file containing several embedded graphics can top a hundred megabytes in size. The result is a big difference in network and file system loading. By 1998, the system's internal operations engine was beginning to break on a routine basis when heavily loaded.

In early 1998, the Imaging Services (CIC-9) staff began working on PAGES99, a project to analyze and overhaul the internal workings of the PAGES system. The existing system had served well, but was no longer efficient or economical to operate. A new system was needed, and it had to be developed and implemented. Resources, staffing, and time were limited—the system had to be replaced before it failed or by the year 2000. Transition to the new system had to be done with a minimal impact on service.

The project was going smoothly and on schedule for a midsummer 1999 switchover when disaster struck in March 1999. An unrecoverable problem developed with the old system. The only way to keep production going was to finish mission-critical parts of the new system on an accelerated basis. This was accomplished by a "big push" of 18-hour days over a 2-week period. PAGES99 has been in full production since that time and to date has processed over 50,000 jobs.

The new system contains 95% less source code, has a new Web page for interactive users, has been moved to smaller and less expensive quarters, and is supported by a smaller staff. At the same time, the new system is more flexible, supports virtual job routing to satellite operations, and is focused on providing new services at lower cost.

How PAGES99 Works

The central key to both PAGES and PAGES99 is the concept that a *job ticket*, which describes a set of *attributes,* is embedded with the job to be processed. The job ticket contains all of the specifications that enable a job to be correctly routed and processed once it reaches its destination.

In the old PAGES system, this information was contained in a fixed, formatted header 360 characters in length at the very beginning of the file. This header record was generated by a program called *ppages,* which enabled a user to specify a job's attributes in a command line. Combining the job ticket as part of the file ensures that the information stays with the file wherever it goes, regardless of what type of method (ftp, email, lpr, etc.), is used to transfer the file from the user's system to PAGES.

PAGES99 uses an expanded header that actually looks just like a UNIX command-line command. The command line is structured in paired sets of [parameter argument] without exception. Thus, instead of the old "-punch" or "-nopunch" argument, the new command line reads "punch [1|0]". The PPAGES99 script compares the command-line arguments with a database table (in text format) that can be readily updated. This means that the source code itself is much more compact and requires much less maintenance. The old ppages source code was roughly 10,000 lines of C. The first-cut *ppages99* source code consists of about 400 lines of Perl.

The command-line syntax is still complex and is recommended only for those users who have batch systems or scripts that generate and transmit jobs automatically. The CIC-9 team is available to assist users in working with command-line syntax.

New Web Page Option

For users who have to send output from their workstations, the best way to get a file to PAGES is via the Web. This interface can be used by Netscape® and Internet Explorer® Web browsers on any type of workstation. The process requires that a *printable file* first be generated (by using the "print to file" option) from the application software. PostScript is the standard file type used. Next, the Web page is used to specify printing details and to send the job to PAGES. Use of the Web page is growing rapidly, with excellent results reported by PAGES customers.

The PAGES99 project has resulted in a tighter, more economical, and more flexible system. Rather than just a printing system, PAGES is now viewed as a *virtual routing* system, sending a job to wherever it can be most appropriately handled. For example, the CIC-9 photo lab now handles large-format plotting. This creates a hub-satellite relationship between the core of the system and various processing locations and even contractors.

Production Process Flexibility

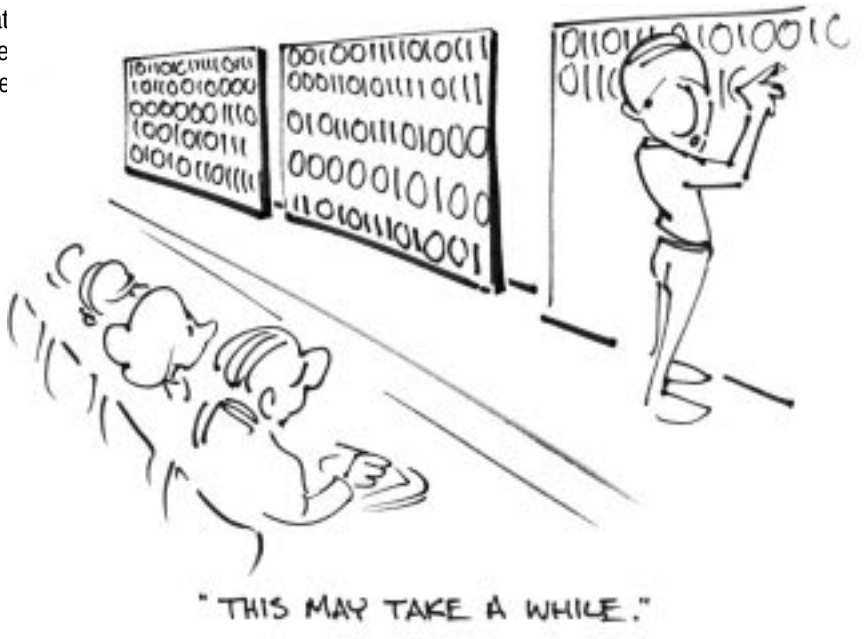
Machine operators now have much more control over the production process because the queuing and device parameters are maintained in a database. The capability of multiple-copy queuing allows operators to mingle on-demand requests along with the regular duty-cycle jobs. Billing for the recharge process is done by using log files from the database.

A Virtual Routing System

Now the system has more capability and flexibility with greater upload and file-size capacity. PAGES99 is device independent, which means that there is a single interface to meet all files types, including .tiff, .text, .ps, and .cgs. As of this writing, .tiff is accepted only on the large-format plotter. The new system is a virtual routing system—like a hub or way station—allowing the system to route the job to the most appropriate output device. For example, photos can be routed to the CIC-9 photography lab.

CIC-9 Group Leader, Henry Johnson, envisions the PAGES99 project as an innovative way to partner with customers. He thinks of PAGES as a place to send all types of imaging requirements—a hub for routing image data of all types. Johnson's vision is that "PAGES is much more than a printing system; ultimately the project will bring people together with solutions for their imaging requirements. Because of the Web, we can offer the advantages of a central service with the convenience of distributed services."

For more information about PAGES99, contact Steve Hench at (505) 667-4034 or send e-mail to shench@lanl.gov. Also, see the PAGES Web site at this URL: http://pages-sv.lanl.gov/ .



Recharge System Telecommunications Enhancements

by Judy Gonzales, Communications Analyst, CIC-4 Billing Team, CIC-4 Telecommunications Group

In September 1999 several new enhancements were added to the Recharge System for CIC-4 Telecommunications. For those employees not familiar with CIC-4—it's the place to call if you need telephones, voice mail, cellulars, calling cards, secure telephones, pagers, radios, network design, external and internal circuits, video teleconferencing, paging systems for buildings, badge readers, video ports, Labnet, and telecommunications recharge/billing information.

Charge codes and names can be changed, as well as requests for disconnects, under the Telecommunications link on the Recharge System Web site (http://recharge/). To use the Recharge System, you must have a token card, for example, a CryptoCard or SecurID. To get a token card, go to this Web site (http://eia/cic.htm) and fill out an ICN Validation Request Form 1646 or e-mail validate@lanl.gov. Employees can make changes to their own services if they have a token card. Managers and Business Team Leaders can make changes to anyone's information in their group/division.

Many people confuse Data Warehouse with the Recharge System. The Recharge System is for maintaining current billing information for future charges and Data Warehouse is for past charges. No changes can be made in Data Warehouse.

In the Recharge System there are two types of changes for telecommunications:

- Standard Charge Code Updates, and
- Global Charge Code Updates.

Standard Charge Code Updates

Enhancement 1: New Service Tab

There are now three criteria tabs to choose from—Customer, Charge Code, and Service. The new Service tab makes it possible to change information by the Service Identifier. For example, the telephone service identifier would be the phone number 7-3347 or 73347. For pagers it would be 104-3347 or l043347, and so on. This is much faster than scrolling through all the lists/pages of numbers and charge codes. Another good feature is the ability to retrieve information without knowing the charge code.

Enhancement 2: New Ways To Sort Information

After a tab is selected, there are several ways to have information sorted.

Customer Tab

Sort by Service Description or Service Identifier

Charge Code Tab

Sort by Charge Code, Name; Name, Service Identifier; Service Identifier or Z number, Service Identifier

Service Tab

Sort by Service Identifier automatically, only one item is retrieved.

Enhancement 3: Short Cut: Return to Criteria

After changes are made and saved, the "Return to Criteria" button at the bottom of the page will take you back to the criteria page so that you can request another retrieval instead of going back to the Recharge System homepage. If you wish to go back to the Recharge System homepage, simply hit the "Done" button if all your changes are completed.

Enhancement 4: "Return to Criteria" Page

The "Return to Open" page was renamed "Return to Criteria."

Global Charge Code Updates

Enhancement 5: Global Updates by Z Number

Changes can now be made by Z numbers in the global table. After the Charge Code information there is a "*From*" Z Number field and a **"***To*" Z Number field (telecommunications only). This kind of global change would transfer all services from Z#A to Z#B. This requires REMGR or REBTL authority.

For detailed instructions on how to use the Recharge System (telecommunications), see the CIC-4 Web site http://int.lanl.gov/orgs/cic/cic4/.

Billing Questions

For questions about billing, call Judy Gonzales at 667-3347 or Alice Martinez at 665-5333, or e-mail cic4_billing@lanl.gov.

Year 2000: Are You Still Ready?

by Denise Sessions, BITS Managing Editor, with Diane Weir and Sue King, Year 2000 Project Office

Lab managers should continue giving priority to ensuring that the Lab is 100 percent ready for the new year. The Lab's goal was to be 100 percent Year 2000-ready by the end of September 1999. By mid-September, 99 percent of the Lab's nearly 23,000 computers and computer systems had been assessed and 93 percent certified as ready for the millennium rollover.

But, even with that much work behind us, just to make sure you're ready, the CIC Year 2000 Project Office recommends that you do another sweep. Even if you were ready 6 months ago, you may not be ready now. Vendors have changed and continue to change their Y2K information.

Check Your Vendor Web Sites Often

Over the past 18 months, hardware and software vendors have dramatically improved Web-based, customer-friendly ways to address Year 2000 issues. The major players now offer a combination of searchable databases to determine product compliance status, user notification-of-changes, free patches and updates, and program-scanning utilities. Most importantly, vendors are still researching and *finding* Year 2000 bugs and quirks, even in GNU Linux freeware. Some vendor diagnostic tools have had second and third revisions. For example, Dell has rev. A01 (replacing rev. A00) of its free Y2000RTC Driver, software that allows you to test and fix your Dell system's real-time clock (RTC). http://www.dell.com/year2000/tools/rtctest.htm

There are so many seemingly trivial new fixes available that it may appear as if vendors have gone overboard in their efforts. Yet the unending changes reveal that Year 2000 is not an easy issue to resolve once and for all. Also, vendors are researching and posting Year 2000 fixes very diligently to avoid future litigation. The *Year 2000 Information and Readiness Disclosure Act* provides legal protection to those manufacturers who openly share Year 2000 information on their products. The vendors have taken that requirement to heart; here are some examples of recent changes:

- Hewlett-Packard *Product Compliance Search* "What's Changed" found 62 changes to HP-UX from January 4,1999 through August 24, 1999. http://www/hp.com/cgi-bin/y2kdb/pSearchpl
- On September 23, IBM broadened and updated its Year 2000 AIX site to *RS/6000 Systems and the Year 2000 issue, including AIX OS and SP System Information and Status*, replacing *AIX, UNIX Operating Systems, and the Year 2000 Issue*. http://www-4.ibm.com/software/year2000/papers/aixy2k.html
- 3Com has a *Year 2000 Product Readiness Disclosure Product Information Updates* Web page http://w3n.3com/cso/y2kweb.nsf/Docum...Product+Information+UpdatesOpendocument, showing 19 changes since July 30, 1999.
- Debian is collecting and postingYear 2000 information on Linux packages and their state of Year 2000 compliance http://www.debian.org/y2k/ and soliciting contributions from users.
- SGI's *IRIX 6.5x Bug Summaries and Workarounds*, last updated on August 31, 1999, lists 7 numbered "Bug References." The site tells customers that "you can achieve Y2K compliance by installing the maintenance releases specified below and/or working around the Y2K bugs as described in the applicable bug reference…" This is only for IRIX 6.5, not for IRIX 5.3, 6.2, 6.3, or 6.4…those updates are promised http://www/sgi.com/tech/year2000/patches.html
- On a Windows NT Year 2000 site last updated September 23, 1999, Microsoft announced in the title that *BIOS Date Value Does Not Immediately Update on January 1, 2000*. See http://support.microsoft.com/support/kb/articles/q2169/13.asp

The lesson to be learned here is to check your vendor Web sites often (registered users may be receiving this information automatically). A quick way to get to vendor Web sites is through the Lab's Year 2000 site, if you don't already have them bookmarked. The URL for the Lab's Year 2000 site is http://www.lanl.gov/projects/ia/year2000/ . The site can also be reached through the Year 2000 link on the Lab internal home page

After all this, even if you think you've done everything you can do, Y2K bugs may still bite. The best way to avoid the bite is to be prepared with a contingency plan and test your systems.

Alternative Ways to Conduct Your Business

As of August 30, 1999, most critical systems at the Lab and all DOE sites have made safety-related and DOE mission-essential systems Year-2000 compliant. Although no one expects any glitches, there are contingency plans in place for those most critical areas of Lab security, safety, and environmental operations. To review the Lab's plan, see the link to the *Year 2000 Business Continuity Plan*: http://int.lanl.gov/projects/ia/year2000/contin/ .

Contingency planning requires you to assume the worst. We don't know if we will have power, if we have found all the bugs, or if vendors are supplying us with comprehensive fixes. We don't know if vendors will continue to give us products and services that we expect from them. Certainly we can act in response to a problem, but we can act more efficiently and avoid additional problems with a contingency plan.

"It appears that all too often the requests on the Internet and elsewhere for "Y2K Compliance" information seems driven by "pointy-haired managers" of the fashion described in Scott Adam's Dilbert cartoon strip."
Christopher Browne's Web Pages
Linux and Year 2000
http://www.ntlug.org/~cbbrowne/linuxy2k.html

What should a contingency plan include? How would you answer these questions?

- What people will you need for operating under unusual conditions?
- What equipment and other resources do you need to support that plan?
- Are your people trained on the contingency plan?
- What are the conditions for invoking the contingency plan?
- Who do you notify?
- How long do you wait before notifying key people?
- What is the outage time; at what point do you do what?
- What is the chain of decision; who do you notify and when?
- Who needs to be there to make the decisions? "People resources" means not only having available the people who fix technical problems, but also key managers who make decisions about how to approach and triage the conditions that exist.
- What do you do if infrastructure around you fails?
- How are you going to communicate in case of a Year 2000 failure? What will you say to whom and how?

Sun's Year 2000 History

Sun Microsystems' Web site has a link called *SunScan 2000 Release History* at http://www.sun.com/y2000/sunscan/history.html. The link provides a summary of feature enhancements and known bug fixes in each version. The latest revision of SunScan is listed first in the "What's New" section.

As an example, Cathy Stallings, CIC-5, has tracked Solaris changes for more than a year. She notes that Solaris version 2.6 was the first version that was intended to be Y2K compliant. Then patches for 2.6 were released. Later, Version 7 was the Y2K-compliant version and now 7 has patches.

A proactive approach to prepare for Year 2000 is to get the latest patch cluster. A nice thing about the patch cluster software is that it checks to see if patches have already been applied. If the patch is not already applied, the software will apply the patch. The patch cluster also checks for software already loaded.

Keep in mind that if you patch now, you may have to apply more patches later and even into the beginning of 2000 as clean-up patches are released. Cathy says to plan on at least 1 hour to install the patch sets. In a test situation at the Lab, the fresh application of all the Y2K and security patches on an Ultra 10, Solaris 2.5.1, took about an hour. Keep an eye out for Return Codes other than 2 or 8. Be sure to read and pay attention to the Return Codes as further action may be required. Be sure you have a lot of disk space before you start making "save sets."

For additional Year 2000 information about Sun Microsystems and Unix, see http://www.lanl.gov/projects/ia/year2000/unix/ .

Consider this example:

Can you operate securely if the kerberos server is down? Under what conditions will you need an exception to operate in the normal security and safety envelope? If an exception is required, does the staff of the agency granting the exception (usually DOE) understand the contingency plan and how they will be involved in its invocation?

For more information about contingency planning, see the Lab Year 2000 Web site's Contingency" section at http://int.lanl.gov/projects/ia/year2000/contin/ .

What To Do Before the Holiday Closure

Be aware of your responsibilities. Assure that the systems you require to perform your job are Year 2000-ready by doing it yourself, or contacting your system administrator, the system owner, or your Year 2000 Council representative. To find out who represents your organization on the Council, see the "Y2K Council" button at this URL http://int.lanl.gov/projects/ia/year2000/council/ .

Things To Do Now

- Develop contingency plans
  - — Design a backup system, including an off-site location, for files and data
  - — Keep in mind that if you plan to back up on ADSM, the system won't handle everyone backing up at the same time at the last minute before the holiday closure
- Test software, systems, and the uninterruptible power supply (UPS)
- Test what vendors say is Y2K compliant

For more information about testing, see the Year 2000 Web site's "Testing" button at http://www.lanl.gov/projects/ia/year2000/test.html .

Things to do before Leaving for the Holiday Closure

These action items are taken from the *Year 2000 Business Continuity Plan*. For more details about preparing for the holiday closure, you can download a copy of the plan from the Year 2000 Web site. You may even want to take with you a copy of this contingency plan and any other plans related to your work.

- Secure all classified material
- Turn off all nonessential electrical loads
  - — Unplug sensitive electronic equipment, including personal computers
- Minimize heating system loads
  - — Close windows and doors
  - — Shut off nonessential exhaust fans to minimize heat loss
  - — Leave thermostats at normal daytime settings
- Fuel government vehicles and emergency generators

CIC Division Year 2000 SWAT Team

The CIC Year 2000 SWAT Team is a supplement to the *Lab-wide Business Continuity Plan* to prepare for possible business and computing interruptions and to ensure rapid recovery following the millennium rollover. The team is charged with developing and implementing a unified CIC Division strategy to deliver customer support in case of computing system failures as a result of Y2K.

Representatives from these organizations comprise the SWAT Team:

- CIC-2: Desktop
- CIC-6: Customer Service
- CIC-13: Business Information Systems
- CIC-15: Database and Information Technologies

The team is responsible for developing the following:

- Year 2000-specific customer management processes,
- prioritization schemes and special procedures for dealing with a high volume of phone calls and increased field work requests, and
- strategies to minimize Y2K impact and costs to the service center.

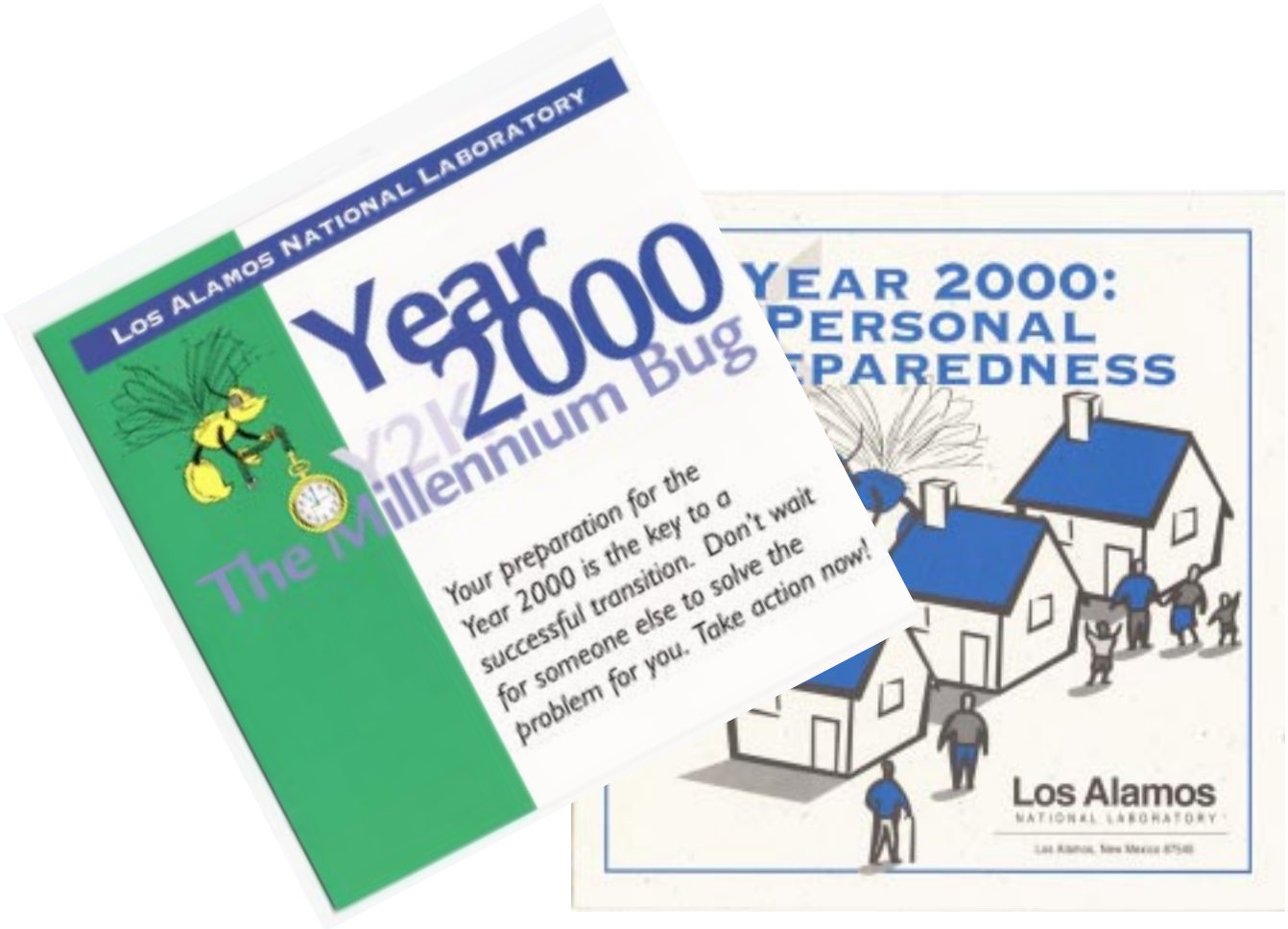
The team is also responsible for training front-line customer service center personnel in Y2K issues. Corissa Yow of the Year 2000 Project Office is the SWAT team contact (yow@lanl.gov).

Other Resources

In addition to the Year 2000 Web site, the Year 2000 Project Office has published these brochures for reference:

- *The Y2K Millennium Bug,* and
- *Year 2000: Personal Preparedness.*

These brochures are also available as .pdf files on the Year 2000 Web site. For specific questions, you may also contact Diane Weir, Year 2000 Project Leader, at (505) 667-9337 or send e-mail to drw@lanl.gov.



Complex Systems Modeling: Using Metaphors from Nature in Simulation and Scientific Models

by Luis Mateus Rocha, Complex Systems Modeling Team, CIC-3, Computer Research and Applications Group (CIC-3)

What are Complex Systems?

"A system comprised of a (usually large) number of (usually strongly) interacting entities, processes, or agents, the understanding of which requires the development, or the use of, new scientific tools, nonlinear models, out of equilibrium descriptions, and computer simulations." (Advances in Complex Systems Journal, italics added.)

"A system that can be analyzed into many components having relatively many relations among them, so that the behavior of each component depends on the behavior of others." (Herbert Simon)

"A system that involves numerous interacting agents whose aggregate behaviors are to be understood. Such aggregate activity is nonlinear, hence it cannot simply be derived from summation of individual components behavior." (Jerome Singer)

A complex system is any system featuring a large number of interacting components (agents, processes, etc.) whose aggregate activity is nonlinear (not derivable from the summations of the activity of individual components) and typically exhibits hierarchical self-organization under selective pressures. This definition applies to systems from a wide array of scientific disciplines. Indeed, the sciences of complexity are necessarily based on interdisciplinary research. Not surprisingly, the heart of this research area is the Santa Fe Institute in Santa Fe, New Mexico, which was founded initially by researchers from the Los Alamos National Laboratory such as George A. Cowan, a LANL Senior Fellow Emeritus. In fact, LANL has produced or attracted many of the researchers associated with this center and research area: Nicholas Metropolis, Chris Langton, Stephanie Forrest, Alan Lapedes, Steen Rassmussen, Alan Perelson, Hans Frauenfelder, among others. However, this research did not recently originate from here; it is a direct offspring of the Cybernetics and Systems Research movements which started in the 1940s with people like Norbert Wiener, Warren McCulloch, Margaret Mead, Ross Ashby, John von Neuman, Heinz von Foerster, and others.[1]

Almost all interesting processes in nature are highly cross linked. In many systems, however, we can distinguish a set of fundamental building blocks, which interact nonlinearly to form compound structures or functions with an identity that requires more explanatory devices than those used to explain the building blocks. This process of emergence of the need for new, complementary, modes of description is known as hierarchical self-organization, and systems that observe this characteristic are defined as complex.[2] Examples of these systems are genetic networks that direct developmental processes, immune networks that preserve the identity of organisms, social insect colonies, neural networks in the brain that produce intelligence and consciousness, ecological networks, social networks comprised of transportation, utilities, and telecommunication systems, as well as economies.

The Complex Systems Modeling Team at the Computer Research and Applications Group (CIC-3) is concerned with basic and applied research on "simulations" of complex systems and development of applications to understand and control such systems. The team's focus is on agent-based simulations of complex systems, such as eco-socio-technical networks, and in the area of biocomputing, with projects in areas such as the study of networks involved in gene expression. Some of this research is described below.

Agent-Based Models

The term agent is used today to mean anything between a mere subroutine and a conscious entity. There are "helper" agents for Web retrieval and computer maintenance, robotic agents to venture into inhospitable environments, agents in an economy, etc. Intuitively, for an object to be referred to as an agent, it must possess some degree of autonomy, that is, it must be in some sense distinguishable from its environment by some kind of spatial, temporal, or functional boundary. It must possess some kind of identity to be identifiable in its environment. To make the definition of agent useful, we often further require that agents must have some autonomy of action, so that they can engage in tasks in an environment without direct external control.

Traditionally, Agent-Based Models (ABM) draw on examples of phenomena from biology such as social insects and immune systems. These systems are distributed collections of interacting entities (agents) that function without a "leader." From simple agents, who interact locally with simple rules of behavior, merely responding befittingly to environmental cues, and not necessarily striving for an overall goal, we observe a synergy which leads to a higher-level whole with much more intricate behavior than the component agents, e.g., insect colonies and immune responses. The field of Artificial Life (AL) [Langton, 1989] produced a number of models based on simple agent rules capable of producing a higher-level identity, such as the flocking behavior of birds, which were called Swarms[3] or ABM. In these models, agents are typically described by state-determined automata: that is, they function by reaction to input and present state using some iterative mapping in a state space. Such ABM can be used, for instance, to simulate massively parallel computing systems, a research interest of several members of our team.

However, it has become clear in recent years, that the modeling of some phenomena, particularly, ecological and social phenomena, requires agents whose behavior is not simply dictated by local, state-determined interaction. In a society empowered by language and hyperlinked by information channels, which in turn impacts planetary ecology, agents have access and rely on accumulated knowledge, which escapes local constraints (via communication), and is stored in media beyond the agent itself and its state. Indeed, many if not most researchers in Artificial Intelligence (AI), Cognitive Science, and Psychology, have come to pursue the idea that intelligence is not solely an autonomous characteristic of agents, but heavily depends on social, linguistic, and organizational knowledge which exists beyond individual agents. Such agents are often known as situated [Clark, 1997] or semiotic [Rocha, 1999] agents. It has also been shown that agent simulations that rely on shared social knowledge can model social choice more effectively [Richards, et al., 1998].

Because most of our research projects deal with modeling social networks, our agent designs need to move beyond state-determined automata with the inclusion of random-access memory capabilities. Our agents are systems capable of engaging with their environments beyond concurrent state-determined interaction by using memory to store descriptions and representations of their environments. They also have access to shared knowledge amongst the members of their particular agent society. Such agents are dynamically incoherent in the sense that their next state or action is not solely dependent on the previous state, but also on some (random-access) stable memory that keeps the same value until it is accessed and does not change with the dynamics of the environment-agent interaction. In this sense, our agent designs create ABM which bridge traditional Artificial Life ABM, AI, and game theory.

One other important characteristic of ABM, which distinguishes them from Cellular Automata (CA), is the asynchrony of the interactions among agents and between agents and their environment. In ABM agents do not simultaneously perform actions at constant time-steps, like CA or Boolean networks. Rather, their actions follow discrete-event cues or a sequential schedule of interactions. The discrete-event setup allows for intergenerational transmission of information, or more generally, the cohabitation of agents with different environmental experience. The sequential schedule setup, formalized by Sequential Dynamical

Systems (SDS) [Barrett and Reidys, 1999], allows the study of different influence patterns among agents, which is very important in the study of decision processes in social networks. The latter are ideal for mathematical treatment as different schedules can be studied in the SDS framework, while the former require statistical experimentation as the collective behavior of discrete-event agents in an environment with stochastic laws and rules cannot be easily studied mathematically.[4]

This way, we can see that ABM, particularly semiotic ABM, draw from Discrete-Event Simulation, AI, Social Science, Complex Systems, and AL. From Discrete-Event Simulation, we require asynchrony, but unlike traditional simulation, we use the distributed ideas of complex systems, requiring that agents do not push for or understand an overall goal, which only emerges from agent interactions. From AI, we require agents with access to memory structures (beyond state-determinacy), and from Social Science, we obtain appropriate game-theoretic frameworks to study agent strategies and decision processes.

Currently, our team is involved in a number of projects using ABM:

Decision Structures of Socio-Technical Organizations. The project aims to analyze the decision structures of complex organizations dependent on information systems for communication and coordination. We develop ABM of such Socio-Technical Organizations (STO), conceived as hybrid systems with an underlying physical system, mediated with a layer of computer-based communication, and interacting with communities of decision-making agents or actors. Target applications include "Operations Management" organizations such as 911/Emergency Response or Search and Rescue operations. The project team includes researchers from the Physical Science Laboratory and the Mathematics Department of the New Mexico State University, Sandia National Laboratory, and Los Alamos National Laboratory.

Agent-based Simulation of Economical and Ecological Systems. Development of ABM of the international economic regime, where the agents are nations that can learn and develop strategies to deal with each other's strategies. Prior, non-agent-based models, have assumed that the world financial system could be modeled as a sequence of static balances, which is plainly false. By contrast, ABM can simulate a more realistic view of the world where oscillations, asset bubbles, and other known phenomena are modeled.

Fundamentals of Simulation LDRD Directed Research. This project advances basic research related to simulation particularly of large socio-technical simulation. The approach relates and illustrates four research areas in simulation science using a telecommunication infrastructure analysis setting. The four areas are: Sequential Dynamical Systems (SDS) [Barrett and Reidys, 1999], which provide theoretical foundations for simulation as ordered composition of dependent local mappings and make connections to algorithm theory; representation for high performance computational implementation of SDS; principles of statistical design and analysis of simulation studies; and foundations for assisted reasoning system design and use. This project includes LANL researchers from TSA-SA, CIC-3, TSA-1, and T-7.

Biocomputing

Biological regulatory mechanisms, including gene expression, are inherently complex systems as defined above. As such, they cannot be understood by mere identification of components, products, ensembles, and connections. Therefore, we wish to supplement the surplus of gene and protein sequence information now available, with informational and developmental means to model and understand evolutionary systems. This includes understanding the representation and communication of information in living systems, predicting protein function from gene sequence information, discovering interactions in gene regulatory networks, and the computer modeling of RNA Editing. We are also considering algorithmic implications of biological problems, such as in the development of efficient DNA sequencing procedures and pooling designs.

Next to the gene sequencing projects, we are now faced with the challenge of mapping a molecular identity–a general gene expression array. Such an array contains the expression intensities of different genes in a cell under certain experimental conditions. We can also construct a matrix of such arrays with information from many cells under the same conditions, or the same cell in different conditions, namely a developmental process or a cell cycle. These matrices contain tremendous amounts of information about the complex interactions between genes, which gene sequencing alone cannot reveal, and ultimately define a cell's behavior.

To discover these patterns of interaction, we are developing pattern recognition algorithms specifically designed to tackle this problem. We are using data-mining techniques such as Association Rules, Fuzzy Clustering, and Mask Analysis (for time-series analysis). Once, this step is finished, we will embark on modeling gene regulation by building network models based on the patterns discovered from the data now becoming available from the genome project. Unlike the social network models described in the first section of this article, here we expect that traditional complex systems

models with state-determined components will suffice (e.g., Boolean or fuzzy networks). Details of this research are available on our Web site.

Related to this research, we are also investigating combinatorial optimization in biology. This research addresses theoretical and practical aspects of optimization problems arising in a biologically-motivated setting. Major efforts include algorithms for optimizing the DNA sequencing process and pooling designs for maximally efficient group testing. (Details also on our Web site.)

Making Use of Other Metaphors from Nature

Our team pursues other related research exploring metaphors from Nature, particularly in the areas of adaptive computation and optimization. The key notion of complex systems, that of many simple processes, under selective pressures, synergistically interacting to produce desirable global behavior, can be applied successfully to different problems. In the area of optimization, we are developing heuristic algorithms inspired by nonequilibrium physical processes. The development of nonequilibrium optimization methods is likely to lead to the next generation of general-purpose algorithms–intended, like simulated annealing, for broad application. We expect that this analysis will lead us to new insights into the role criticality plays in combinatorial optimization, as well as to a deeper (and more applied) understanding of computational complexity.

Similarly, we are developing biologically motivated designs for Adaptive Knowledge Management. Distributed designs that draw from immune system metaphors and other aspects of biological systems can largely improve existing information retrieval and knowledge management in networked information resources. We are developing a system for LANL's Research Library that allows different databases to learn new and adapt existing keywords to the categories recognized by different communities, using algorithms inspired by biological and cultural evolution.

For more information e-mail: rocha@lanl.gov or rocha@santafe.edu or see this URL: http://www.c3.lanl.gov/~rocha .

Footnotes

[1]For more details on Cybernetics and Systems Research please refer to Klir [1991] or the web pages of the Principia Cybernetica Project.

[2]Hierarchical self-organization and emergence have been discussed in detail by Pattee[1978], Rosen [1993], and Cariani [1992].

[3]This research led to the development of an agent-based simulation language called Swarm, which we use in several of our projects.

[4]More details of our agent designs are available in our web site.

References

Barrett, C. L. and C. M. Reidys, "Elements of a Theory of Computer Simulation: Sequential CA Over Random Graphs," *Applied Mathematics and Computation* **98**, 241–259 (1999).

Cariani, Peter, "Emergence and Artificial Life," in *Artificial Life II*, C. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, Eds, *SFI Series in the Sciences of Complexity*. (Addison-Wesley, Reading, MA, 1992) pp. 775–797.

Clark, Andy, *Being There: Putting Brain, Body, and World Together Again* (The MIT Press, Cambridge, MA, 1997).

Klir, George J., *Facets of Systems Science* (Plenum Publishing Corp., New York, 1991).

Langton, C. G., "Artificial Life," in *Artificial Life*, C. Langton, Ed. (Addison-Wesley, Reading, MA, 1989).

Pattee, Howard H., "The Complementary Principle in Biological and Social Structures," *Journal of Social and Biological Structures* **1**, 191–200 (1978).

Richards, D., B. D. McKay, and W. A. Richards, "Collective Choice and Mutual Knowledge Structures," *Advances in Complex Systems* **1**, 221–236 (1998).

Rocha, Luis M., *Syntactic Autonomy: or Why There is No Autonomy without Symbols and How Self-Organizing Systems Might Evolve Them*. (New York Academy of Sciences, New York, in press). (Not available yet as the paper is in press, but see the paper at this Web site: http://www.c3.lanl.gov/~rocha/sa2.html .)

Rosen, Robert, "Bionics revisited," in *The Machine as a Metaphor and Tool*, H. Haken, A. Karlqvist, and U. Svedin, Eds. (Springer-Verlag, New York, 1993), pp. 87–100.

Dimensionless Coding Techniques in Tecolote

Using a Single Source-Code Base for Multidimension Programs

by John H. Hall, Technical Staff Member, X-10: Code Group B

Dimensionless programming techniques replace explicit [i][j][k] notation with containers that are traversed by iterators. I begin by discussing the more general issue of what is a container and what is an iterator, followed by demonstrating some of the algorithms for building the offset lists that you use to construct these iterators. The last part of the article shows examples of these techniques implemented using POOMA Release 1 and Tecolote version 2.3.1.Most of the examples in this article feature the dimension-independent concepts introduced by POOMA. A future article will highlight additional techniques used within the Tecolote framework.

Containers

The Standard Template Language (STL) specification states that containers are

"*objects that store other objects. They control allocation and deallocation of these objects through constructors, destructors, insert, and erase operations.*"

Because the most common form of container, the vector, has the overloaded operator[], you may use the [i][j][k] syntax, but you lose the ability to write dimension-independent code. If you use iterators instead, you can write code that—at setup time— loops over the appropriate dimension and fills the container. Usage of this iterator does not require knowledge of the dimension.

If you are careful not to rely on the specific traversal path of the iterator, you can write a single source code that will correctly handle any dimension. Using this same technique, you can iterate over a cell's neighbors in either a Lagrangian or an Eulerian code, which means that you can write a dimension- *and* mesh-independent code.

The Tecolote framework offers many containers to the programmer, including those provided by the STL (such as vector, list, deque, and map) as well as the POOMA framework's Index, NDIndex, and Field containers. Tecolote typically uses the STL's vector class along with the POOMA containers to achieve its dimension independence.

The current version of Tecolote can initialize STL vectors directly from the input file, and therefore Tecolote commonly uses these vectors to initialize POOMA's containers. Direct initialization of POOMA's containers is not yet possible because no corresponding Tecolote metatype has been written for them, and they do not have all the methods necessary to mimic an STL container. Future versions of Tecolote will be able to initialize POOMA containers directly from the input file.

POOMA Containers and the Tecolote Input File

The POOMA NDIndex container is a collection of POOMA Index objects that each define a single dimension's range by storing a begin index, end index, and stride. Many constructors are defined for the index object, but the most general one supplies three integers that represent the begin, end, and stride. The NDIndex container, accessed at run time, then has a complete description of a dimension-dependent range—allowing you to write the source code in a dimension-independent manner.

The sample code below demonstrates how to create these containers in a dimension-independent manner:

```
const int Dim=3;

int begin=0, end=100,stride=1;

NDIndex<Dim> allCells;

for(int d = 0; d < dim; d++) {

allCells[ d]  = Index(begin,end,stride);

}
```

In this example, the only parameters that might need to be initialized as persistents in the input file are begin, end, and stride. You can obtain the begin, end, and stride information in three ways: (1) use an STL vector, (2) simply declare three persistents in the parent class, or (3) get the parameters from the Mesh class (except for stride, which is usually 1 anyway). You should become accustomed to seeing the *for* loop over dimension: it will appear in the setup almost any time you try to use a container that is hiding dimension-dependent code. The example above initializes the NDIndex to the same range in each dimension.

Using the NDIndex is remarkably simple. The POOMA Field has overloaded the operator[] to simply take an NDIndex. Unfortunately, if one field in an expression uses an NDIndex operator, all other fields in that same expression will require one. The first example below is clearly acceptable because both fields have the same NDIndex range. In the second

example, the Density Field does not have an NDIndex, so not all Fields in the same expression are properly indexed and the compilation will fail.

```
Density[ allCells]  = OLDDensity[ allCells]  * 0.5;  OK

Density = OLDDensity[ allCells]  * 0.5;          Bad

Pressure[ allCells]  = n* R* Temperature[ allVerts]
 Volume;  Bad
```

In the last example, there are two problems. (1) The Volume field does not have an NDIndex object, whereas both Pressure and Temperature do. (2) The Temperature Field and the Pressure Field have different NDIndex containers and, although this is legal, all NDIndex objects in a single expression must be conformant; that is, they must contain exactly the same number of elements in each dimension. (This second point is a potential problem that is discoverable only at run time.)

Another POOMA container that we often want to initialize from the Tecolote input file is the Vektor<double,dim> class. Unfortunately, this container does not have exactly the same semantics as STL's vector<double> class. Specifically, it does not have the push_back and erase methods. Consequently, you must have a vector<REAL> persistent class member for initialization, and its values must be copied to the POOMA Vektor<REAL,dim>.

```
class A {
public:
PERSISTENT_MEMBERS(A)
.
.
.
virtual void initialize(void);
private:
vector<REAL> initXYZ;
 Vektor<REAL,Dim> XYZ;
};
 BEGIN_PERSISTENT( A )
 PERSISTENT( VectorOfReals, intXYZ, intXYZ )
END_PERSISTENT


void A::initialize() {
// Remember me? The dimension loop?
for(int d = 0; d < Dim; d++)
XYZ[ d]  = initXYZ[ d] ;
}
```

This extra initialization step is annoying and it is targeted for death in a future release of Tecolote. However, the source code is completely capable of working in any dimension without using the grotesque #define DIM == 1, etc., preprocessor monstrosity. Also, because this duplication happens only once during initialization, there is no impact on performance.

Neighborhood Iteration Using NDIndex Containers

The POOMA Field operator[] can accept more than just an NDIndex container. By supplying an integer array of size Dim for offsets in addition to the NDIndex container, you can perform stencil operations. This is essentially a translation of the block described by the NDIndex.

An example of this usage is Tecolote's VectorVelocity command, which was written to initialize each cell in the Velocity field to a value whenever that cell or any of its neighbors has a non-zero volume fraction. This one routine demonstrates three useful tricks for dimension-independent coding: (1) initializing a Vektor from a vector through the input file, (2) using offsets with NDIndex objects, and (3) summing over the volume fractions in each dimension to determine whether the cell or any of its neighbors have non-zero values. (Note that we do not care which cell is non-zero, just that one of them is.)

```
class VectorVelocity : public Command {
public:
 PERSISTENT_MEMBERS(VectorVelocity)
 .
 .
 .
//  Methods
 void initialize( void );
private:
 DataDirectory *dd;
 vector<REAL> Velocities;
};


BEGIN_PERSISTENT( VectorVelocity )
 PERSISTENT( vector<REAL>, Velocities, Velocities)
END_PERSISTENT
void VectorVelocity::initialize(void)
{
```

```
  int d1,d2;
  int off[ DIM][ DIM] ;
  Vektor<REAL,DIM> InitialVelocity;

  // It's me again, copying the values from the
  Input File
  for(d1 = 0;d1 < DIM;d1++)
  InitialVelocity[ d1]  = Velocities[ d1] ;

  // Get Tecolote mesh object and other necessary
  fields

 PhysicsMesh& Mesh =
  ( STRICT_GET( "Mesh", *dd,  PhysicsMesh ) );
  VECTOR_FIELD(Vert)& Velocity =
  STRICT_GET( "Velocity", *dd, VECTOR_FIELD(Vert) );
  SCALAR_FIELD(Cell)& VolFrac =
  STRICT_GET( "VolFrac", *dd, SCALAR_FIELD(Cell) );

 // ------------------------------
 // Set constant velocity for the material passed in
 // ------------------------------

 // Build DIM x DIM 2D Array of Offsets
 for(d1 = 0;d1<DIM;d1++) {
  for(d2 = 0; d2<DIM;d2++) {
  off[ d1][ d2]  = 0;
  }
    off[ d1][ d1]  = -1;
  }

 SCALAR_FIELD(Vert) SumVof(Mesh);
 const NDIndex<DIM>& dom = SumVof.getDomain();
 SumVof[ dom]  = VolFrac[ dom] +VolFrac[ dom+off[ 0]] ;

 for(d1 = 1;d1<DIM;d1++)
 SumVof[ dom]  = SumVof[ dom] +SumVof[ dom+off[ d1]] ;

  Velocity = where(gt(SumVof,EPSILON),
 InitialVelocity, Velocity);
}
```

First, you initialize an STL vector<REAL> from the Input File. The

```
vector<REAL> Velocities;
```

line in the declaration sets up the storage for your STL container. The

```
BEGIN_PERSISTENT( VectorVelocity )
PERSISTENT( vector<REAL>, Velocities, Velocities )
END_PERSISTENT
```

lines register your vector as a persistent data member of the VectorVelocity class.

Second, copy the STL vector's contents into the POOMA Vektor :

```
for(d1 = 0;d1 < DIM;d1++)
InitialVelocity[ d1]  = Velocities[ d1] ;
```

After getting your fields out of the DataDirectory, set up your offsets. Because you are not trying to identify the specific cell that is nonzero, all you have to do is sum up the volume fractions from the cell of interest and its neighbors.

Visualizing the algorithm requires one to realize the Velocity as a vertex-centered quantity, whereas the VolFrac is cell-centered. Consequently, there are 2, 4, or 8 neighbors for 1D, 2D, or 3D, respectively.

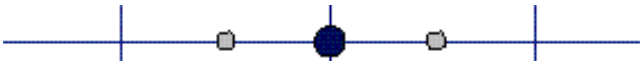


Fig. 1. 1D diagram summing cell-centered volume fractions along X-axis and placing the result at the vertex.

Start by initializing the vertex-centered field `SumVof` with the sum of the two cells from the cell-centered VolFrac field along the X-direction (Figs. 1 and 2).

```
SumVof[ dom]  = VolFrac[ dom] +VolFrac[ dom+off[ 0]] ;
```

In 1D, you are finished; the next loop will be skipped. For 2D

and 3D, you continue to add the vertices along the Y-direction. At this point you have the sum of the cells along the X-Y plane.

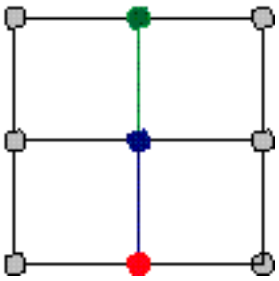


Fig. 2. The vertex-centered values in the field SumVof along the blue and green lines are summed; their result is placed at the vertex of the same color (blue and red go to blue, green and blue go to green)

Finally, for the 3D case only, you repeat the summation along the Z-direction. Each vertex now contains the sum of the eight surrounding cell-centered volume fractions.

```
for(d1 = 1;d1<DIM;d1++)

SumVof[ dom]  = SumVof[ dom] +SumVof[ dom+off[ d1]] ;
```

The final line initializes the Velocity field to InitialVelocity whenever any neighbor has a nonzero volume fraction; otherwise, it simply replaces the original value.

```
Velocity = where(gt(SumVof,EPSILON),

InitialVelocity, Velocity);
```

It may seem odd to have to replicate the Velocity field in the false position for the where statement. However, when you consider how expression templates work, the reason for this oddity becomes clearer. Consider the case in which we pass the result of a where statement as a parameter to a function:

```
foo(where(gt(SumVof,EPSILON),InitialVelocity));
```

We can see that the *where* expression has no way to know which field is receiving the result. Therefore, it is impossible to predict the value when the expression evaluates to false.

Summary

The simple examples above give the flavor of the techniques used to write dimension-independent code. In practice, you must recognize patterns that are far more complex if you want to fully achieve dimension independence. Once you discover such a pattern, the result is often beautiful and almost self-describing. In fact, when the correct centerings and patterns are used, you can sometimes compress the lines of written code for an expression by a factor of 100. In addition, finding the right pattern significantly reduces errors due to coding; once debugged in 1D form, the 2D and 3D versions are almost always correct. A future BITS article that uses more elegant examples will amply demonstrate these benefits.

Book Review

Philip and Alex's Guide to Web Publishing*, by Philip Greenspun, Academic Press/Morgan Kaufmann, April 1999*

Reviewed by Chris Lindberg, CIC-1, Communication Arts & Services

I laughed, I cried . . . Philip Greenspun's engaging anecdotes brought a smile each time they rang true with my own experiences as a Web developer and a sigh and a frown whenever I realized the mistake he was describing was one of my own.

Philip and Alex's Guide to Web Publishing

Some would question the wisdom of dual publishing a book in paper and on the Web at the same time. (http://www.photo.net/wtr/thebook/)

Personally I have found it to be a great boon. I was able to read most of the book in paper on a trip to the Midwest. No need to lug along my laptop, no eyestrain, and no batteries required. Having a copy on the Web has made it easy to search for and retrieve information (and my favorite quotes) to share them with friends and colleagues.

Rarely have I found a book that is both entertaining and full of good technical information and insights. Not only have I learned a great deal about building truly dynamic Web sites using technically sound hardware and software solutions, I have enjoyed reading a good book.

My Favorite Quote:

"Computers are tools of the devil. It is as simple as that. There is no monotheism strong enough that it cannot be shaken by Unix or any Microsoft product. The devil is real. He lives inside C programs."

New Interface for Biosciences Database

by Jeane Strub, Electronic Databases Team Member, CIC-14, Research Library

A new interface and search engine is now available for BIOSIS® at LANL, our version of the world's leading database for the life sciences. The new system uses Verity's Topic search engine and is similar to SciSearch at LANL. Features include:

- Alerts—weekly e-mail delivery of new information for searches of your choice
- Easy limiting by date, language, document type, and electronic format
- Phrase searching
- Sorting of results by relevancy ranking, date, author, or title
- Ability to search individual author names with initials
- Links to both print and electronic research library holdings
- Ability to Mark All for printing, e-mail, or downloading

The BIOSIS database has 12 million records covering literature from 1969 to the present and is updated bi-monthly. BIOSIS indexes approximately 6500 journals and 2000 meetings/year, as well as books and other materials. Subject coverage includes biological and biomedical sciences, botany, biochemistry, biophysics, biotechnology, medicine, public health, radiation biology, ecology, and the environment.

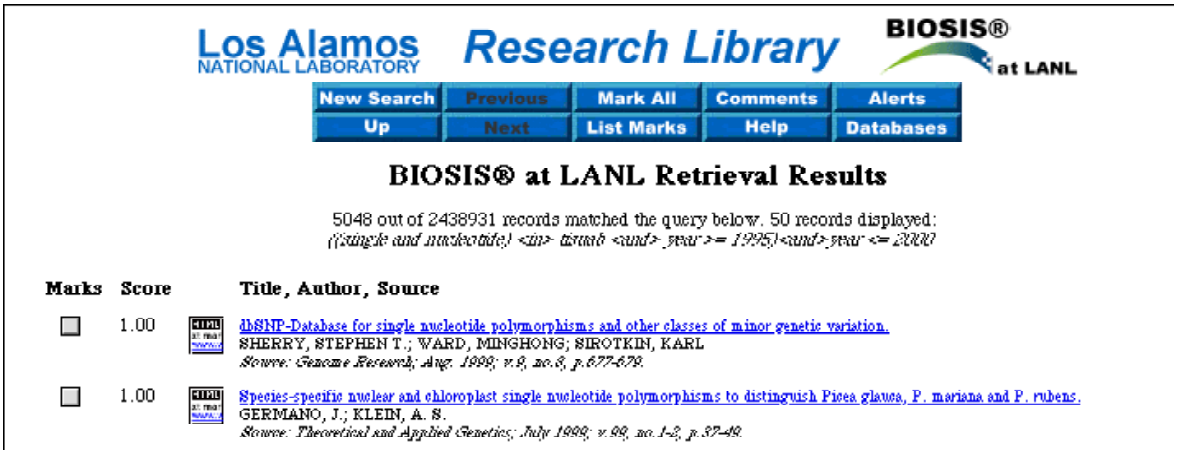
Beilstein Organic Chemistry Database Is Now on the Web—Free

by Lou Pray, Customer Service Team Member, CIC-14, Research Library

Frustrated by the lack of Web-based Chemistry research sources? Beilstein, the leading organic chemistry reference source, is now available free of charge on ChemWeb.com. Users are able to access titles, abstracts, and citations from the top journals in organic and related chemistry, published from 1980 to the present. Currently there are approximately 600,000 articles in the Beilstein Abstracts Database.

You may search by author or by words included in the title or abstract of the paper. You may also restrict your search to a certain journal or time range, and a table of titles is delivered for your inspection free of charge. You can then view all detailed information by choosing the "Full Record" hyperlink. The full record is displayed in the following format: Title, Author, Reference, CNR, Abstract.

Access it at this URL: http://chemweb.com/databases/beilstein .



Sample BIOSIS® Search results.

LOS ALAMOS NATIONAL LABORATORY

Research Library

http://lib-www.lanl.gov

The LANL Research Library offers a variety of training opportunities for the Laboratory community. Available sessions focus on specialized library databases and other electronic resources. A complete list of course offerings can be found at http://lib-www.lanl.gov/libinfo/training.htm. All sessions are available to individuals or groups, at the library or your site. Arrange for a session by contacting the Library, phone 7-4175 or e-mail library@lanl.gov. Library tours are available on a drop-in basis every Wednesday at 1:00 p.m.

Attention Application Developers!!!

Need User Guides?

Need Online Help?

CIC-1's Software Documentation Team can help you improve your product by developing user manuals, online help, quick reference cards, or other documentation to help your users learn the application. Contact Sheila Molony, 5-1585, or sheila.molony@lanl.gov, for more information.

Technical and Advanced Technical Computer Training Courses

Communications	Office Skills 2000	Web Authoring and Browsing
• Eudora 4.x • Meeting Maker 5.0.3	• Office Skills 2000–LANL • Office Skills 2000–Professional Development	• Dreamweaver 2.0–MAC or PC • FrontPage 2000 • HTML Basics • HTML Intermediate
Enterprise Information Applications (EIA)	**Other EIA Courses**	**System Administration Training**
• Date Warehouse–Basics • Date Warehouse–EDS Reports • Data Warehouse–Passport • EDS–Basics • EDS–GUI • EDS–Training Plans • Infomaker • Invoice Approval System • Key/Core System • Purchase Card System • Procurement Desktop • Recharge • Time & Effort GUI • Travel Foreign GUI • Travel Domestic GUI • Web JIT	• Financial Management Information System (FMIS) • Property Accounting, Inventory, and Reporting System (PAIRS) • Signature Authority System (SAS) • Secretarial/Contract Service (SE) • Salary Review System (SRS) • Directory Information System (DIS) • Automated Chemical Information System	• IRIX (SGI) System Administration (Beginning) • IRIX (SGI) System Administration (Advanced) • IRIX (SGI) Network Administration • SGI Performance Evaluation and System Tuning • Solaris 7 System Administration • Solaris 7 Network Administration • Solaris 7 Server Administration • Unix and Widows NT Integration • Windows NT Workstation and Server • Windows NT Optimization and Troubleshooting • Windows NT Security
Programming Training	**Application Training**	**ASCI**
• C Programming (Beginning) • C Programming (Advanced) • C++ for Experienced C Programmers • ANSI/ISO C++ Programming Clinic (Advanced C++) • Distributed Objects Using Corba • Java Programming • Java Program Workshop • Distributed Programming with Java • Programming for Beginners Using Java • Object Technology– A Management Overview • Object-Oriented Analysis and Design • Perl Programming • Advanced Perl Programming with CGI • C-Shell Programming • Python for Scientists and Engineers	• Advanced WWW Development • FrameMaker Basic and Advanced • Foundations of IDL Programming • IDL 5.0 Graphic Object Workshop • Mastering MFC Development Using Visual C++ 6 • Netscape Servers for Intranet Development • Origin 2000 Applications Programming and Optimization • Sendmail–Managing Internet Mail • C++ and the Unified Modeling Language • Sybase Fast Track to Adaptive Server Enterprise 11.5 (ASE) • Sybase Performance and Tuning for System 11 • Sybase SQL Server Administration • Unix (Beginning) • Unix (Advanced) • Visual Basic 6.0 Fundamentals • Visual C++ Windows Programming • Windows NT Security • Windows 2000 Training	• Getting Started on ASCI Blue Mountain Systems • Running MPI on Blue Mountain Systems • Introduction to Totalview • LSF (Load Sharing Facility) • Introduction to HPSS (High Performance Storage System)

*You do not need an ICN password to use e-mail.

Reader Feedback

Feedback helps us to provide a document that responds to the changing needs of its readership. If you have comments or questions about this publication, please let us hear from you. We have reserved the back of this form for that purpose. We also accept articles for publication that are of interest to our readers. Contact the managing editor for more information. This form is also used for new subscriptions, deletions, or changes. Instructions are on the back. If you prefer to contact us by e-mail, send your comments and/or subscription request to denise@lanl.gov.

Do Not Staple
Fold on This Line First

cut along dashed line

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 88 LOS ALAMOS NM

POSTAGE WILL BE PAID BY THE ADDRESSEE

MAIL STOP B251
ATTN: DENISE SESSIONS, MANAGING EDITOR
CUSTOMER SERVICE GROUP (CIC-6)
LOS ALAMOS NATIONAL LABORATORY
PO BOX 1663
LOS ALAMOS, NM 87545-9916

Do Not Staple, Seal with Tape
Fold Here

1999 12-Month Index

To electronically access any of the articles listed on this index go to http://www.lanl.gov/orgs/cic/cic6/bits/index/index_home.html. Note that the page numbers printed on this page correspond to the page numbers in the printed versions of the publication.

Keywords	Title of BITS Article	Date	Page
ASCI	*Blue Mountain Is World's Fastest Computer*	*Dec. 98*	*1*
	Tecolote: An Object-Oriented Framework for Hydrodynamics Physics	*Aug. 99*	*26*
	Using Tecolote Components to Extend Object-Oriented Programming	*Sept. 99*	*19*
ASCI Training	*ASCI Training Course Descriptions*	*J/J 99*	*27*
BITS Announcements	*BITS Gets New Design, New Editor*	*F/M 99*	*3*
BITS Interviews	*Bob Newell*	*O/N 98*	*1*
	Earleen Eden	*O/N 98*	*3*
	Dale Land	*Dec. 98*	*11*
	John Morrison	*F/M 99*	*6*
	Ann Hayes	*J/J 99*	*22*
	CIC Division Hosts Student Poster Presentations	*Sept. 99*	*23*
CIC (Computing, Information, & Communications)	*BITS Welcomes New Deputy Division Director for Strategic Computing*	*F/M 99*	*6*
Desktop Computing	*CIC-2's Future Is in Stewardship, Automation, and "Fun Stuff"*	*Dec. 98*	*11*
	New Ways for Managing Software Development Projects	*J/J 99*	*2*
	Scientific Software Process Engineering	*J/J 99*	*4*
	Adjusting for Reality: Mitigating Uncertainty in Projects	*Aug. 99*	*7*
	Your Data Is Gone . . . Now What?	*Aug. 99*	*11*
	Desktop on Demand: A Desktop That Displays on Your Computer, but Actually Runs on the Enterprise Server	*Sept. 99*	*6*
	Grassroots Software Management: Simple Things to Do	*Sept. 99*	*8*
	News About Microsoft Networking at Los Alamos: The LANL Master Accounts Domain Is Up and Running!	*Sept. 99*	*10*
	Notes on Setting Up and Configuring an NT Domain	*Sept. 99*	*12*
	A Primer for Developing Online Help	*Oct. 99*	*4*
	News about Laboratory Electronic Mail and Policies	*Oct. 99*	*6*
Embedded Systems	*The Year 2000 Bug Is Hiding*	*F/M 99*	*1*
ESD	*Working for and Serving You*	*F/M 99*	*10*
HTML	*Web Content Architecture: Taming the Tangle of Protocols*	*Sept. 99*	*15*
	Extensible HTML	*J/J 99*	*12*
	MathML: A Kind of "TeX for the Web"	*Oct. 99*	*2*
High Performance Computing	*Performance and Scalability Analysis of Applications on T-flop-scale Architectures*	*J/J 99*	*7*
	Major 3-D Parallel Simulations	*J/J 99*	*9*

Keywords	Title of BITS Article	Date	Page
High Performance Computing (Cont.)	*Performance and Scalability Analysis of Applications on T-flop-scale Architectures*	*J/J 99*	*7*
	Major 3-D Parallel Simulations	*J/J 99*	*9*
	Lecture Review: Next-Generation Chips, Processors, Transistors, and Wiring	*Aug. 99*	*23*
	Tecolote: An Object-Oriented Framework for Hydrodynamics Physics	*Aug. 99*	*26*
	Frameworks Are Models, Too!	*Oct. 99*	*16*
	Automated OS Install HOWTO: Linux Cluster via Gigabit Ethernet	*Oct. 99*	*20*
Infrastructure	*Virtual Private Network*	*J/J 99*	*15*
	Electronic Information Protection Regimes	*Aug. 99*	*13*
	Web Content Accessibility: New W3C Guidelines Have Arrived	*Aug. 99*	*20*
	Web Content Architecture: Taming the Tangle of Protocols	*Sept. 99*	*15*
	Data Analysis in the Advanced Surveillance Technology Initiative: AMISS Testbed	*Oct. 99*	*9*
	Transaction Integrity and Validity in MC&A Databases	*Oct. 99*	*12*
Information Systems	*Engineering Index Is Now Available from the Research Library*	*Dec. 98*	*13*
	Research Library Adds Social SciSearch® Database	*F/M 99*	*5*
	Library Databases	*J/J 99*	*20*
	aha!—An Interface to the Best of LANL Science & Technology Web Resources	*Aug. 99*	*2*
	More Than a New Look & Feel: The New LANL Badging System	*Aug. 99*	*5*
	Laboratory Overview Presentation Materials on the Web	*Sept. 99*	*2*
	New INSPEC® at LANL	*Sept. 99*	*28*
	Keep on Top of Information Technology with Gartner Group	*Sept. 99*	*28*
Networks	*Laboratory Unclassified Network Will Implement Changes for Security*	*Dec. 98*	*3*
	Preserving Bandwidth on Your Windows NT Network	*F/M 99*	*8*
	News About Networking at Los Alamos: The LANL Master Accounts Domain Is Up and Running!	*Sept. 99*	*10*
	Notes on Setting Up and Configuring an NT Domain	*Sept. 99*	*12*
SciSearch	*Research Library Adds Social SciSearch® Database*	*F/M 99*	*5*
Supercomputing	*Blue Mountain Is World's Fastest Computer*	*Dec. 98*	*1*
Virtual Private Network	*Virtual Private Network*	*J/J 99*	*15*
Wildfires	*Forecasting Wildfires and Other Crises*	*Apr. 98*	*3*
Windows NT	*Preserving Bandwith on Your Windows NT Network*	*F/M 99*	*8*
	News about Networking at Los Alamos: The LANL Master Accounts Domain Is Up and Running!	*Sept. 99*	*10*
	Notes on Setting Up and Configuring an NT Domain	*Sept. 99*	*12*
World Wide Web (WWW or Web)	*Lab Web Publishers Invited to IntraLab99*	*F/M 99*	*7*
	Extensible HTML	*J/J 99*	*1*
Writing Resources	*Telling Your Story in BITS: Sharing Your Expertise and Energy*	*Aug. 99*	*31*
Year 2000 (Y2K)	*The Year 2000 Bug Is Hiding*	*F/M 99*	*1*

Customer Support Center . . . . (505) 665-4444, ext. 851, or cichelp@lanl.gov

Because of a wide variety of CIC computing services, numerous facilities are available to address your questions. If you are uncertain whom to call, you can always call the Customer Support Center (CSC). CSC consultants are trained to either answer your question or locate someone who can. To reach the appropriate consultant, dial 665-4444 and make your selection from the following choices:

Option 1: New user topics including e-mail, passwords, registration, and World Wide Web
Option 2: Enterprise Information Applications such as Travel, Time and Effort, and Purchase cards
Option 3: Scientific computing, storage systems, and networking
Option 4: Classroom instruction and training
Option 5: Desktop Consulting for PC and Macintosh software and network configurations.

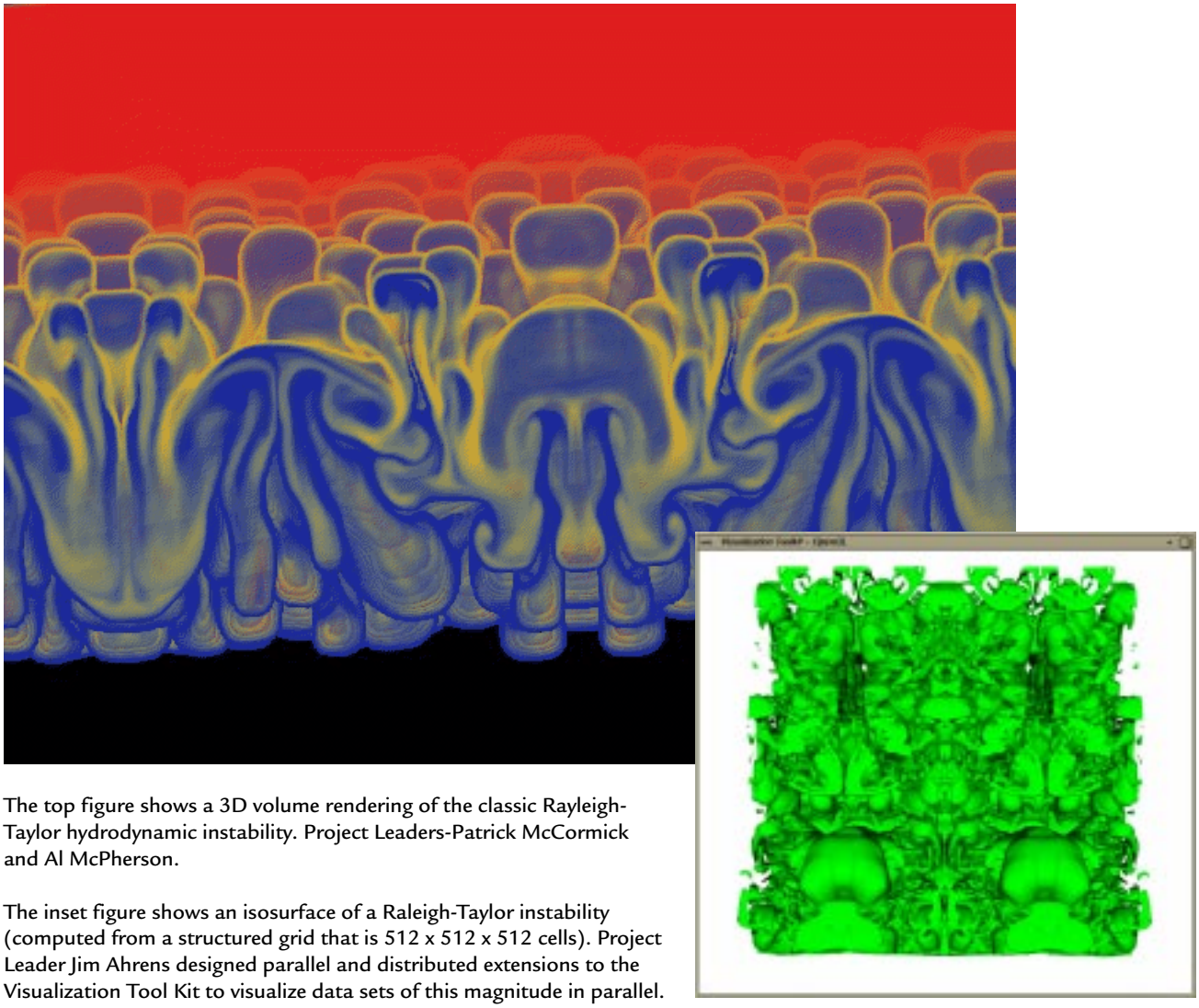
Consulting Via E-Mail

Customer Support ..... cichelp@lanl.gov
Scientific and engineering computing..... consult@lanl.gov
Administrative and business computing .... eiaconsult@lanl.gov
Passwords and registration ..... validate@lanl.gov
Macintosh computing ..... Mac-help@lanl.gov
PC computing ..... PC-help@lanl.gov
UNIX computing ..... UNIX-help@lanl.gov

Other Useful Numbers

Advanced Computing Laboratory 665-4530
Central Computing Facility 667-4584
Network Operations Center ..... noc@lanl.gov or 667-7423
Telephone Services Center 667-3400

Los Alamos
NATIONAL LABORATORY



The top figure shows a 3D volume rendering of the classic Rayleigh-Taylor hydrodynamic instability. Project Leaders-Patrick McCormick and Al McPherson.

The inset figure shows an isosurface of a Raleigh-Taylor instability (computed from a structured grid that is 512 x 512 x 512 cells). Project Leader Jim Ahrens designed parallel and distributed extensions to the Visualization Tool Kit to visualize data sets of this magnitude in parallel.

Both visualization tools shown here and the figure on the inside front cover were created by the visualization research scientists at the Advanced Computing Laboratory. For more information e-mail the Visualization Team at viz@acl.lanl.gov.

Los Alamos
NATIONAL LABORATORY

Los Alamos, New Mexico 87545

A U.S. Department of Energy Laboratory

BITS is published monthly to highlight recent computing and communications activities within the Laboratory. We welcome your suggestions and contributions.

BITS may be accessed electronically at this URL:
http://www.lanl.gov/orgs/cic/computingatlanl/

LALP-99-42 (11/99)